XCWM:
The X11 Composite Window Manager Library

Jess VanDerwalker

Background

- Portland State University Capstone project
  - Original team:

    Ben Carr, Ben Huddle, Aaron Skomra,

    Braden Wooley, Jess VanDerwalker
  - Sponsors:

    Jamey Sharp

    Josh Triplett   - XCB development

    Jeremy Huddleston Sequoia - XQuartz

XCWM Goals

- API that allows display of X11 clients in non-X environments

- Increase code reuse and improve maintainability

- Lead to deprecation of specific DDX's in server code base

- XCB as interface to X protocol

- DAMAGE and Composite extensions

XtoQ
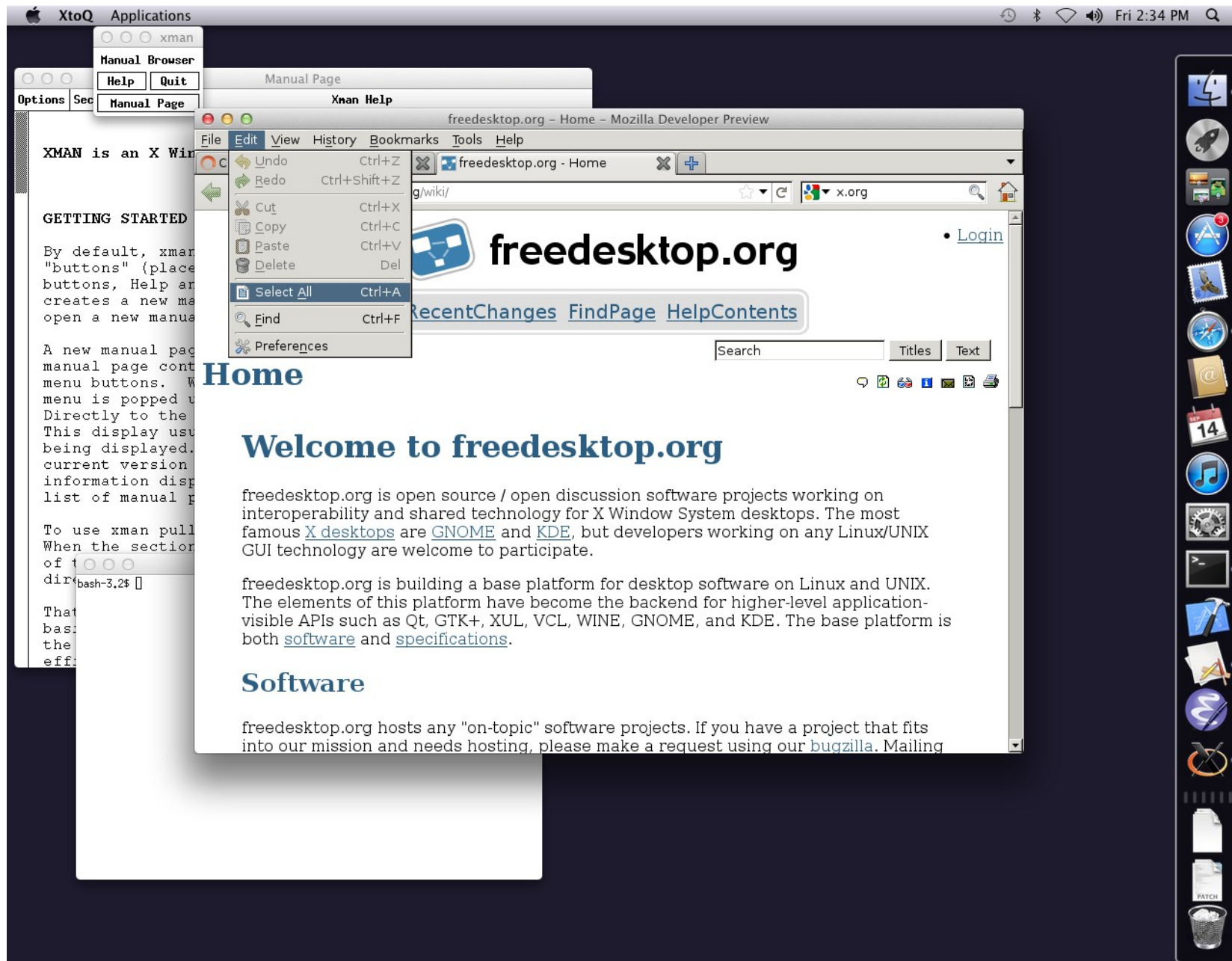
Where we are...

- Basic window manager functionality
- Support for a subset of ICCCM and EWMH protocols
- Functional mouse and keyboard input
- Support for override-redirect windows

Basic functionality

XCWM:

- Create and maintain data structures

- Catch and process X events, pass to WM code

- Catch and process WM events, pass to X
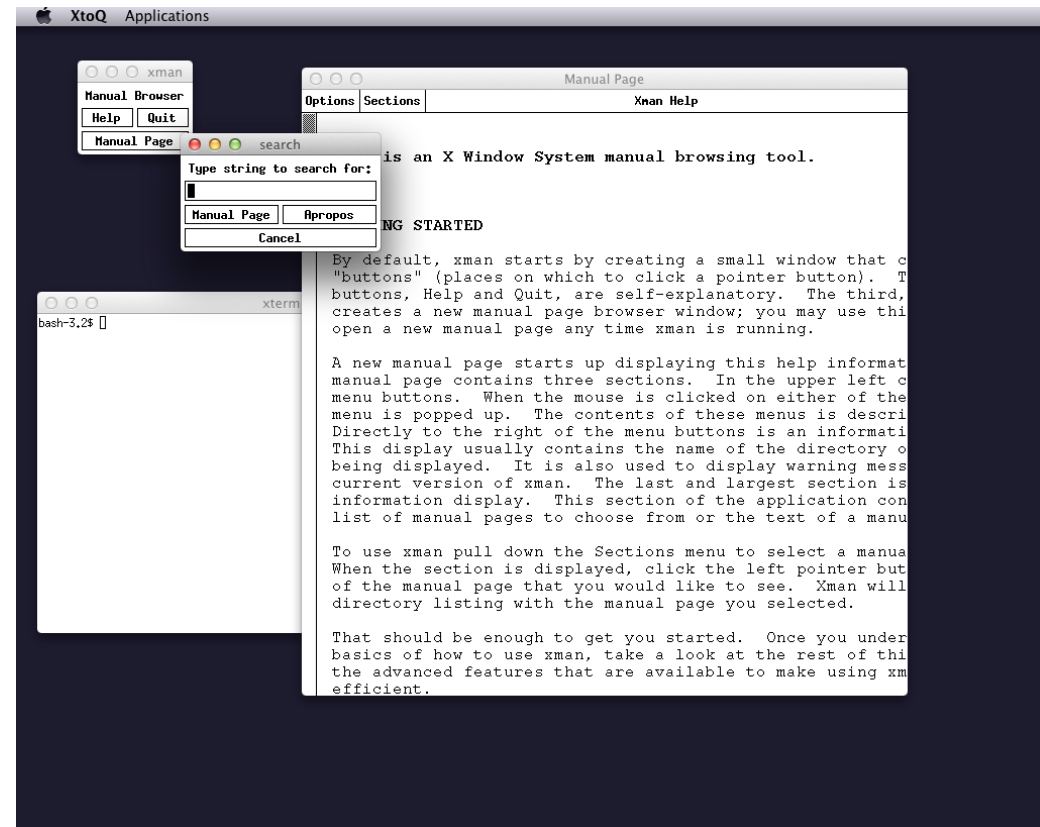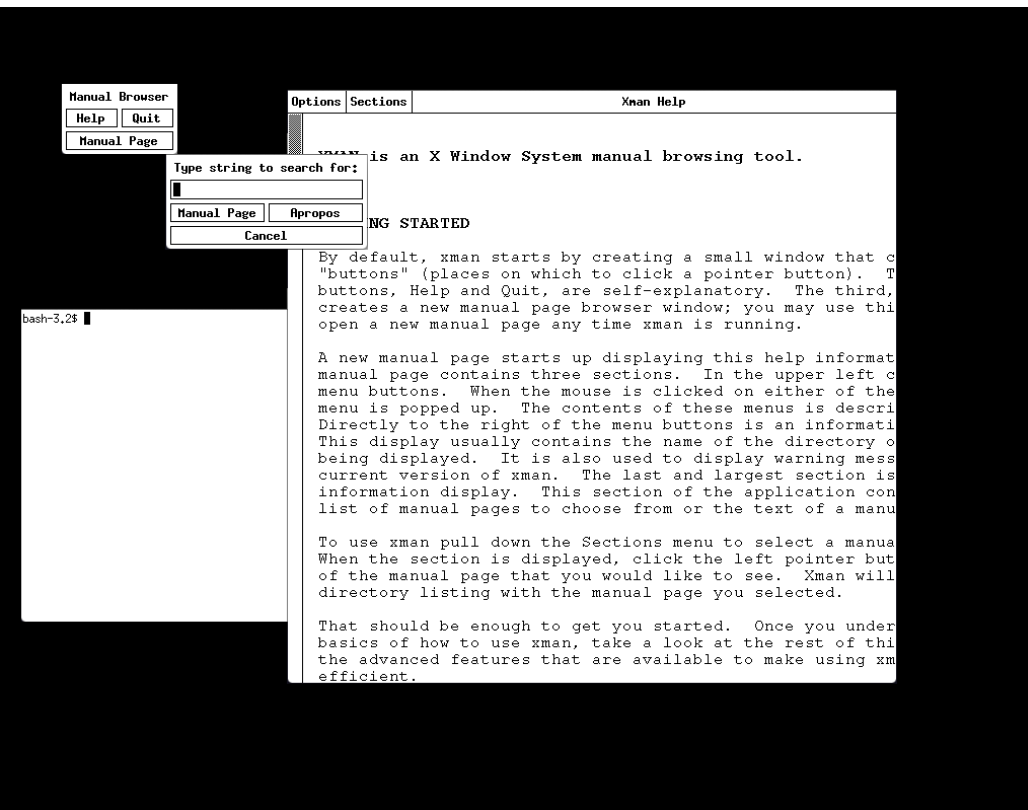
- Provide access to data

Window Manager:

- Render X client pixmaps with native decorations

- Respond to XCWM events, update windows accordingly

- Pass local window events to XCWM

Basic window management

dummy driver

XtoQ

Event handling

From X:

- while loop with xcb_wait_for_event() running in separate thread

- Events passed to WM through WM provided callback function

From WM:

- Passed to X through XCWM function calls

- WM provides asynchronous dispatching if necessary

Client window mapped

XCB_MAP_NOTIFY received from server → Determine if window is being managed

Query attributes of window

Send window created event through callback

size
position
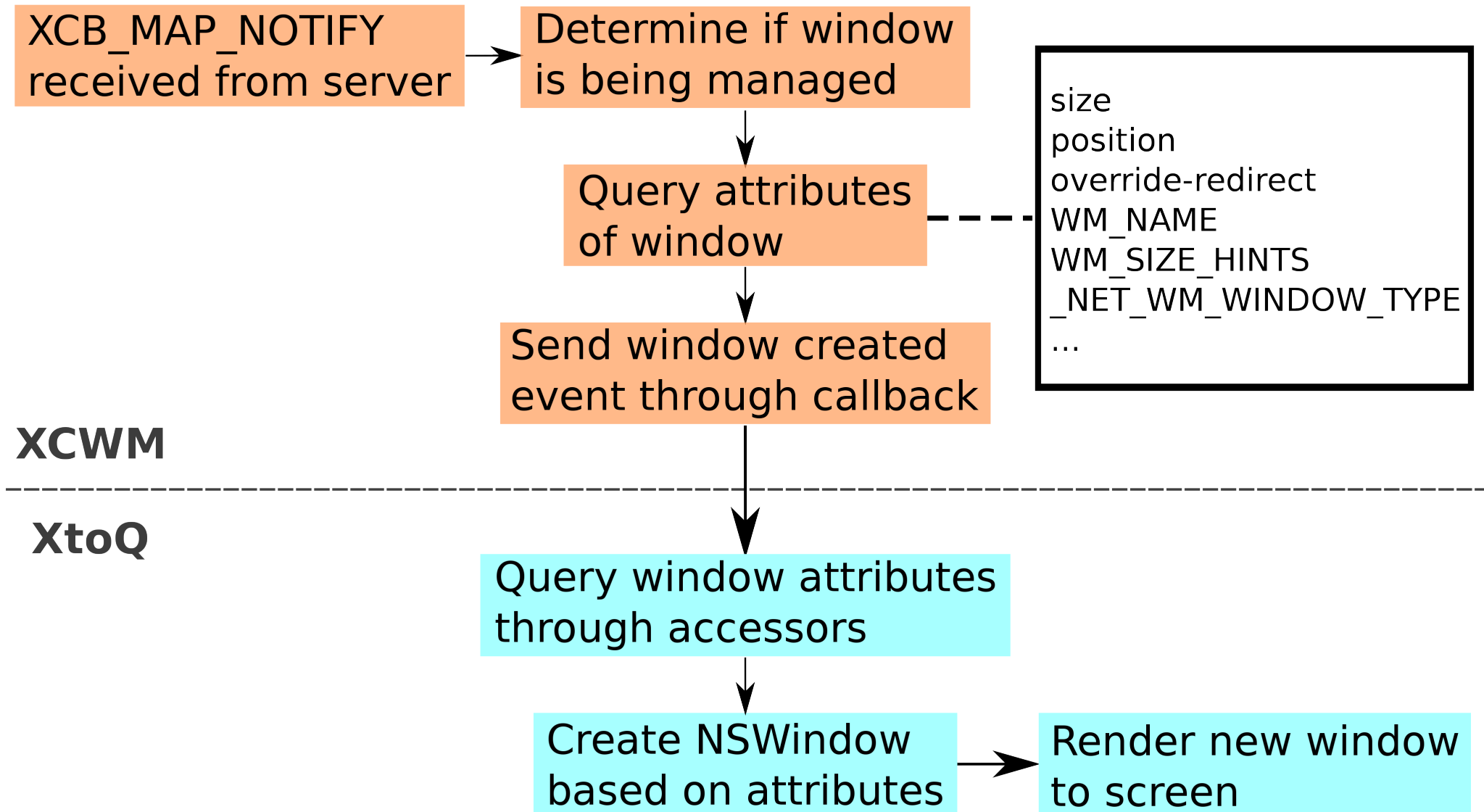override-redirect
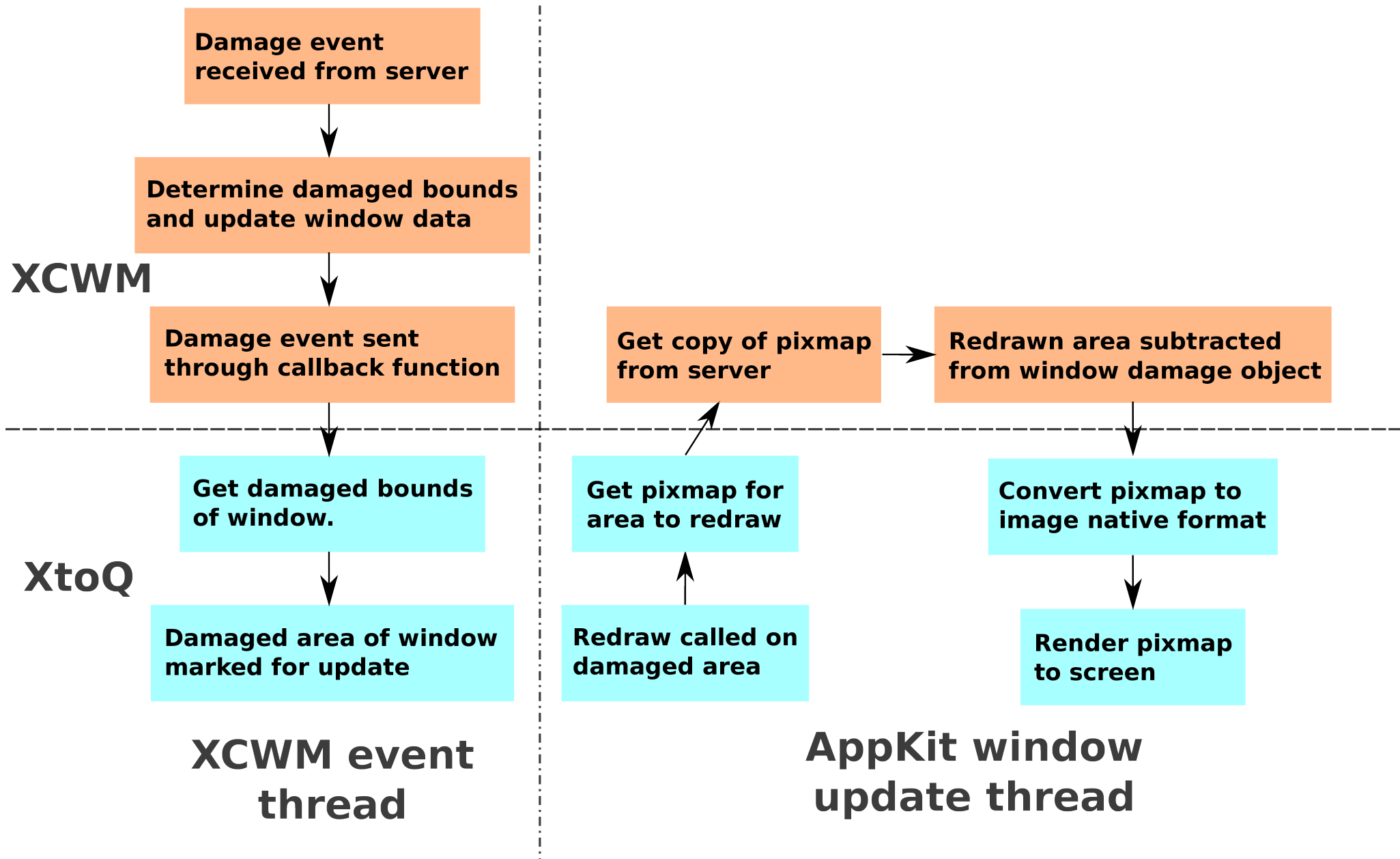WM_NAME
WM_SIZE_HINTS
_NET_WM_WINDOW_TYPE
...

XCWM

XtoQ

Query window attributes through accessors

Create NSWindow based on attributes → Render new window to screen

Damage event and window update path

Damage event received from server

↓

Determine damaged bounds and update window data

↓

XCWM

Damage event sent through callback function

Get copy of pixmap from server → **Redrawn area subtracted from window damage object**

↓

Get damaged bounds of window.

Get pixmap for area to redraw

Convert pixmap to image native format

XtoQ

↓

↑

↓

Damaged area of window marked for update

Redraw called on damaged area

Render pixmap to screen

XCWM event thread

AppKit window update thread

Keyboard and mouse input

Input event caught by AppKit → Event type determined

Event type determined → XCWM handler called using asynchronous dispatch

XtoQ

- -

XCWM

XTEST call to inject input event

Server
Determine input destination client from focused window / mouse position
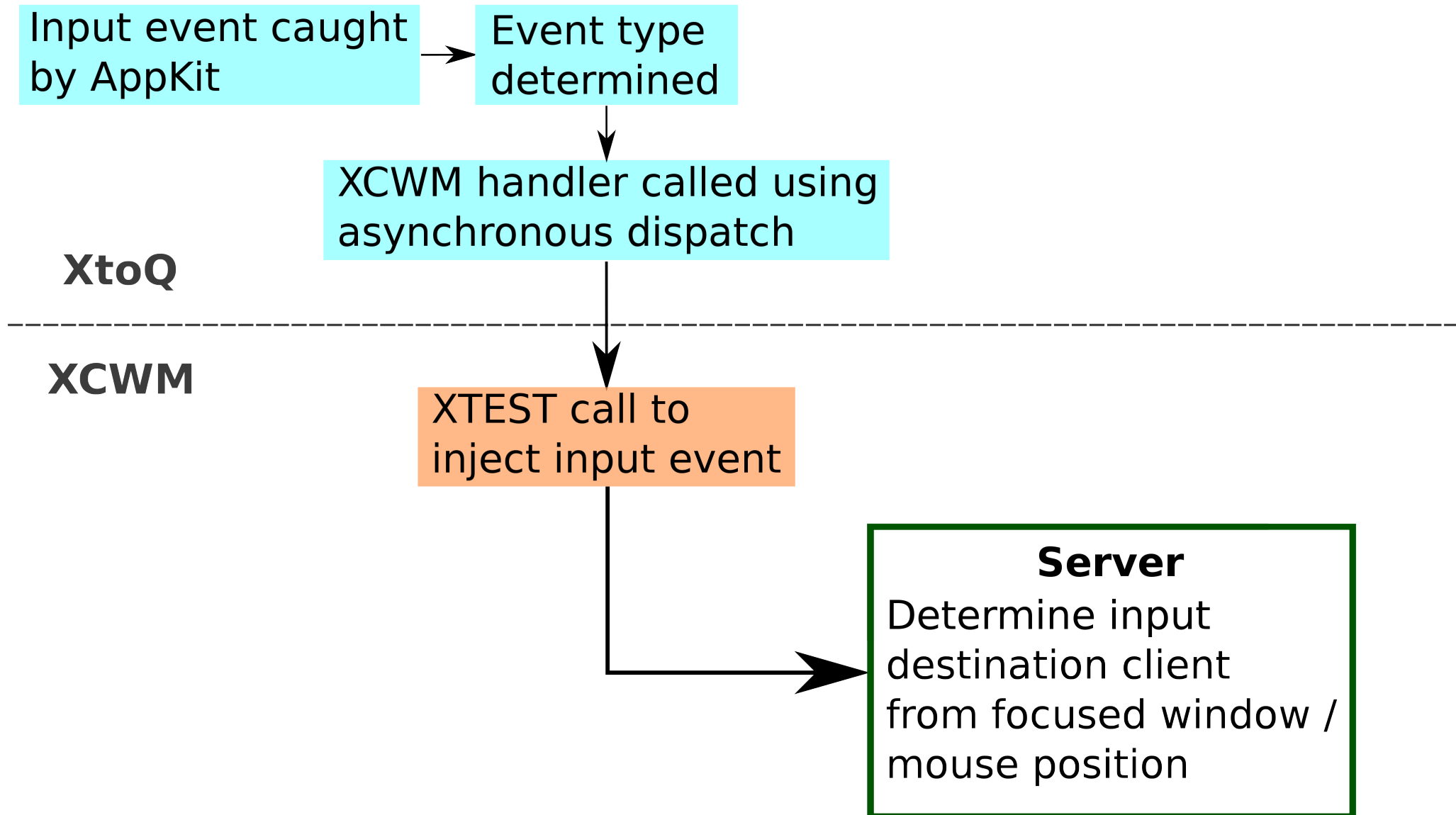
Future work

- More ICCCM / EWMH protocol support
- Rethink event loop
- XtoSomethingElse
- Replace XTEST

Want to contribute?

XCWM git repository:

http://cgit.freedesktop.org/xorg/lib/libxcwm/

Contact:

Jess VanDerwalker: jvanderw@freedesktop.org

washu@sonic.net